

GS Java - Projeto Sustentavel

Autores:

Nicolas Lorenzo Ferreira da Silva - RM 557962

Pedro Henrique Faim dos Santos – RM557440

Disciplina: Engenharia de Software - Domain Driven Design - Java

Professor: Gilberto Alexandre das Neves

RESUMO

Este projeto implementa uma API RESTful em Spring Boot com persistência em Oracle, front-end básico com Thymeleaf e validações nos DTOs. O objetivo é demonstrar domínio de Domain Driven Design (DDD) aplicado a um cenário de projetos voltados à sustentabilidade.

ESPECIFICAÇÃO

Justificativa: ferramenta para gerenciamento de projetos sustentáveis na universidade.

Objetivos: CRUD de projetos, gerenciamento de recursos e usuários, validações, deploy em Render.

DER

(Diagrama Entidade-Relacionamento) - resumo textual:

Tabelas principais: USUARIOS, PROJETOS, RECURSOS

USUARIOS(ID, NOME, EMAIL, TIPO)

PROJETOS(ID, TITULO, DESCRICAO, STATUS, DATA_INICIO, USUARIO_ID(FK))

RECURSOS(ID, TIPO, QUANTIDADE, PROJETO_ID(FK))

DIAGRAMA CLASSES

Diagrama de Classes: O diagrama está disponível em docs/diagrama_classes.puml (PlantUML). Para gerar a imagem localmente: use o PlantUML (plantuml.jar) ou <https://plantuml.com/pt/servlet>. Comando local: `java -jar plantuml.jar diagrama_classes.puml`
-> gera PNG

PROT TELAS

Protótipos de Tela:

- index.html: listagem de projetos
 - projeto-form.html: formulário de criação/edição
 - projeto-details.html: detalhes do projeto
- (Templates básicos em src/main/resources/templates/)

ENDPOINTS

Tabela de Endpoints:

GET /api/projetos -> Lista

GET /api/projetos/{id} -> Detalhe

POST /api/projetos -> Cria (body: ProjetoCreateDTO)

PUT /api/projetos/{id} -> Atualiza

DELETE /api/projetos/{id} -> Remove

(Replicar para /api/usuarios e /api/recursos)

VALIDAÇÕES

DTOs usam Jakarta Bean Validation (@NotBlank, @Size, etc.).

Controllers recebem @Valid e GlobalExceptionHandler trata
MethodArgumentNotValidException.

DEPLOY

Deploy (instruções resumidas):

1. Criar repositório no GitHub e dar push do código.

2. Em Render, criar novo Web Service, conectar ao repositório.

3. Adicionar variáveis de ambiente:

SPRING_DATASOURCE_URL=jdbc:oracle:thin:@:/ <HOST>:<PORT>/<SERVICE>

SPRING_DATASOURCE_USERNAME=<USER>

SPRING_DATASOURCE_PASSWORD=<PASS>

4. Se Oracle bloquear conexões externas, peça liberação do IP do Render ou utilize um banco alternativo.

OBSERVAÇÕES

Oracle JDBC driver pode não estar disponível no Maven Central; confirme se necessita upload manual.

Commits: garanta que o commit final esteja antes do prazo. Incluí um commits.txt com exemplos